We Claim:

1. A method of maintaining consistency of content of an object and metadata related to said object in a loose transaction model for object and meta-data updates, said method including the steps of:

storing said related meta-data and a reference to said object in a table of a database, said object being stored externally to said database in an object store, said reference used to obtain a handle for directly accessing or manipulating said external object;

obtaining a version number embedded in said handle; and

comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object.

2. The method according to claim 1, further including the steps of:

if said encoded version number and said version number of said latest committed version match, comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and

if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object.

3. The method according to claim 2, further including the step of, if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object.

4. The method according to claim 1, further including the steps of updating said object in-place under either DBMS control or file system control and linking said meta-data and said object under DBMS control.

5. The method according to claim 1, wherein said loose-transaction update model uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database.

6. The method according to claim 1, further including the step of intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object.

7. The method according to claim 6, wherein said intercepting step is carried out using a filter layer of said object store for said stored object.

8. The method according to claim 1, wherein said object store is a local file system.

9. The method according to claim 2, wherein said object store is a distributed file system, said object being accessed from a remote file system client.

10. The method according to claim 9, wherein a file access occurs in the presence of authoritative caching and said comparing steps are performed at saidfile system client.

11. The method according to claim 10, further including the steps of caching the last known version number and the corresponding last modification timestamp at said file system client after an access and refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time one or both of said comparisons fail with the previously cached values, in which case said comparing steps are retried with refreshed values.

12. The method according to claim 1, wherein said object includes a file.

13. The method according to claim 1, wherein said version number associated with said object is embedded in an access token.

14. The method according to claim 1, wherein said version number is temporally unique.

15. The method according to claim 13, wherein the last-modification-timestamp attribute associated with said object is maintained by said object store.

16. The method according to claim 1, wherein clock synchronization between a database server and a filesystem server is not required.

17. The method according to claim 1, wherein said database is rolled back to an earlier state.

18. The method according to claim 1, wherein said database is a replicated version.

19. The method according to claim 1, further including the steps of:
updating said object while said object is currently linked; and
accessing said meta-data for said object while said object is being updated.

20. An apparatus for maintaining consistency of content of an object and metadata related to said object in a loose transaction model for object and meta-data updates, said apparatus including:
means for storing said related meta-data and a reference to said object in a table of a database, said object being stored externally to said database in an object store, said reference used to obtain a handle for directly accessing or manipulating said external object;
means for obtaining a version number embedded in said handle; and
means for comparing said embedded version number with a version number of a

latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object.

21. The apparatus according to claim 20, further including:

means for, if said encoded version number and said version number of said latest committed version match, comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and

means for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object.

22. The apparatus according to claim 21, further including means for, if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object.

23. The apparatus according to claim 20, further including means for updating said object in-place under either DBMS control or file system control and linking meta-data and said object under DBMS control.

24. The apparatus according to claim 20, wherein said loose-transaction update model uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database.

25. The apparatus according to claim 20, further including means for intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object.

26. The apparatus according to claim 25, wherein said intercepting means uses a filter layer of said object store for said stored object.

27. The apparatus according to claim 20, wherein said object store is a local file system.

28. The apparatus according to claim 21, wherein said object store is a distributed file system, said object being accessed from a remote file system client.

29. The apparatus according to claim 28, wherein a file access occurs in the presence of authoritative caching and both said means for comparing are implemented at said file system client.

30. The apparatus according to claim 29, further including means for caching the last known version number and the corresponding last modification timestamp at said file system client after an access and means for refreshing said last known version number and said corresponding last modification timestamp with latest values from a file server the next time with previously cached values, in which case both comparing means retry said comparisons with refreshed values.

31. The apparatus according to claim 20, wherein said object includes a file.

32. The apparatus according to claim 20, wherein said version number associated with said object is embedded in an access token.

33. The apparatus according to claim 20, wherein said version number is temporally unique.

34. The apparatus according to claim 33, wherein a last-modification-timestamp attribute is associated with said object and maintained by said object store.

35. The apparatus according to claim 20, wherein clock synchronization between a database server and a filesystem server is not required.

36. The apparatus according to claim 20, wherein said database is rolled back to an earlier state.

37. The apparatus according to claim 20, wherein said database is a replicated version.

38. The apparatus according to claim 20, further including:

means for updating said object while said object is currently linked; and

means for accessing said meta-data for said object while said object is being updated.

39. A computer program for maintaining consistency of content of an object and metadata related to said object in a loose transaction model for object and meta-data updates, said computer program including:

computer code for storing said related meta-data and a reference to said object in a table of a database, said object being stored externally to said database in an object store, said reference used to obtain a handle for directly accessing or manipulating said external object;

computer code for obtaining a version number embedded in said handle; and

computer code for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object.

40. The computer program according to claim 39, further including:

computer code for, if said encoded version number and said version number of said latest committed version match, comparing a last modification time stamp of said object with a last modification timestamp for said latest committed version of said object; and

computer code for, if said last modification time stamp of said object matches with said last modification timestamp for said latest committed version of said object, permitting access to said externally stored object.

41. The computer program according to claim 40, further including computer code for, if said last modification time stamp of said object does not match with said last modification timestamp for said latest committed version of said object, generating an error to indicate that said handle refers to stale content in said object.

42. The computer program according to claim 39, further including computer code for updating said object in-place under either DBMS control or file system control and linking said meta-data and said object under DBMS control.

43. The computer program according to claim 39, wherein said loose-transaction update model uses SQL Mediated Object Manipulation (SMOM) for an object that resides external to said database.

44. The computer program according to claim 39, further including computer code for intercepting a native access to said externally stored object or a file system and validating the caller's access rights based on a combination of said version number and a last modification timestamp for a version of said object.

45. The computer program according to claim 43, wherein said intercepting computer code uses a filter layer of said object store for said stored object.

46. The computer program according to claim 39, wherein said object store is a local file system.

47. The computer program according to claim 39, wherein said object store is a distributed file system, said object being accessed from a remote file system client.

48. The computer program according to claim 46, wherein a file access occurs in the presence of authoritative caching and both said computer codes for comparing are carried out at said file system client.

49. The computer program according to claim 39, wherein said object includes a file.

50. The computer program according to claim 39, wherein said version number associated with said object is embedded in an access token.

51. The computer program according to claim 39, wherein said version number is temporally unique.

52. The computer program according to claim 49, wherein a last-modification-timestamp attribute is associated with said object and maintained by said object store.

53. The computer program according to claim 39, wherein clock synchronization between a database server and a filesystem server is not required.

54. The computer program according to claim 39, wherein said database is rolled back to an earlier state.

55. The computer program according to claim 39, wherein said database is a replicated version.

56. The method according to claim 39, further including:

means for updating said object while said object is currently linked; and

means for accessing said meta-data for said object while said object is being updated.

57. A computer program product having a computer readable medium having a computer program recorded therein for maintaining consistency of content of an object and metadata related to said object in a loose transaction model for object and meta-data

updates, said computer program product including:

computer program code means for storing said related meta-data and a reference to said object in a table of a database, said object being stored externally to said database in an object store, said reference used to obtain a handle for directly accessing or manipulating said external object;

computer program code means for obtaining a version number embedded in said handle; and

computer program code means for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object.

58. A system for maintaining consistency of content of an object and metadata related to said object in a loose transaction model for object and meta-data updates, said system including:

a database storing said related meta-data and a reference to said object in a table of a database, said reference used to obtain a handle for directly accessing or manipulating said object;

a native object store for storing said object externally to said database;

a database mediator for obtaining said handle using said reference to directly access or manipulate said external object;

means for obtaining a version number embedded in said handle; and

means for comparing said embedded version number with a version number of a latest committed version of said externally stored object to determine if said handle refers to a current version of said externally stored object.